7

CHAPTER

Vox Populi: Evolutionary Computation for Music Evolution

Artemis Moroni Technological Center for Informatics
Jonatas Manzolli University of Campinas
Fernando Von Zuben University of Campinas
Ricardo Gudwin University of Campinas

As I cannot come to you at present, I am in the meantime addressing you using a most excellent method of finding an unknown melody, recently given to us by God and I found it most useful in practice. Further, I most reverently salute Dom Martin, the Prior of the Holy Congregation, our greatest helper, and with the most earnest entreaties I commend my miserable self to his prayers, and I admonish Brother Peter, who, nourished by our milk, now feeds on the rudest barley, and after golden bowls of wine, drinks a mixture of vinegar, to remember one who remembers him.

. . .

To find an unknown melody, most blessed brother, the first and common procedure is this. You sound on the monochord the letters belonging to each neume, and by listening you will be able to learn the melody as if from hearing it sung by a teacher. But this procedure is childish, good indeed for beginners, but very bad for pupils who have made some progress. For I have seen many keen witted philosophers who had sought out not merely Italian, but French, German, and even Greek teachers for the study of this art, but who, because they relied on this procedure alone, could never become, I will not say skilled musicians, but even choristers, nor could they duplicate the performance of our choir boys.

—*Guido d'Arezzo (Strunk 1950)*

7.1 INTRODUCTION

Systems for algorithmic composition evolved side by side with the arising of Western music. One of the first known proposals to formalize composition was made by the Italian monk Guido d'Arezzo, in 1026, who resorted to using a number of simple rules that mapped liturgical texts in Gregorian[1] chant, due to an overwhelming number of orders for his compositions. In the classical era, composers such as Mozart, Haydn, and C. P. E. Bach used an algorithmic decision process called "Würfelspiel" (Dice Game) to compose minuets and other works. The music was constructed by means of random selection of segments from a table of motifs. Not surprisingly, in the 20th century, John Cage used Tarot and I Ching to build musical architectures (Loy 1988).

The development of electrical and electronic devices brought electronic musical instruments to the musical realm. Nowadays, the computer represents a technological tool to study music in a way that was not possible in the past. As stated by Moore (1990):

> Computers allow precise, repeatable experimentation with sound. In effect, musicians can now design sounds according to the needs of their music rather than relying on a relatively small number of traditional instruments . . . New devices extend the capabilities of musicians to control the production of sound during live performances. Whatever else they may represent, computers are the most flexible and most powerful instruments of music yet devised.

Several approaches to composing music with the aid of a computer have been developed since Max Mathews's early research (1963), which brought to sonic composition the computer as a new musical instrument, or the composition of the ILLIAC suite by means of stochastic processes (Hiller and Baker 1964). In the last four decades, much research has brought about many computer music systems, with models derived from nonlinear dynamics (Bidlack 1992; Manzolli 1993). Other methods have explored formal grammars and their extensions, sometimes in the direction of cellular automata (Miranda 1994, and Chapter 6).

1. The designation "Gregorian" refers to Pope Gregory I, who ruled from 590 to 604, and who is generally believed to have played a decisive role in the final arrangement of the chants, each of which he (or rather those to whom he had entrusted the task) assigned to a specific occasion of the liturgical year, according to a broadly conceived plan (Apel 1958).

Musical composition can be seen as a framework to express human subjectivity in symbolic and subsymbolic terms. Such a field is exactly the type of domain in which an interactive genetic algorithm (IGA) is most useful. An application of genetic algorithms to generate jazz solos has been described in Chapter 5, and this technique has also been studied as a way of controlling rhythmic structures (Horovitz 1994). While recent techniques of digital sound synthesis bring a large number of new sounds to the musician's desktop, several artificial intelligence techniques have been applied to algorithmic composition. On the one hand, a new sound imagery; on the other hand, several approaches attempt to organize a wide panoply of new sounds:

> The general area of compositional algorithms reflects the tremendous variation in approach to music making which has been the hallmark of contemporary music for decades (Moore 1990).

In this chapter we introduce a new system, Vox Populi, based on evolutionary computation, for composing music in real time. A population of chords is properly codified according to the MIDI protocol and evolves by the application of genetic algorithms. A fitness criterion is defined to indicate the best chord in each generation, and this chord is selected as the next element in the sequence to be played. Each new generated chord is a new sound palette that a musician can use to continue the music evolution. Graphic controls (pad and sliders) provide user-friendly manipulation of the fitness and of the sound attributes. Evolutionary computation is used to stimulate the user with novel sounds, and it allows the user to respond.

Associating the dynamic behavior of genetic algorithms with these tools for real-time interaction, Vox Populi becomes a musical instrument. But unlike a traditional instrument, Vox Populi is able to create its own sound raw material (chord population) and to provide choice criteria (music fitness) simultaneously. All these features enhance the user's music capabilities and mark this system as the state of the art in computer music.

Next, a general description of the main components of the computational environment and melodic, harmonic, and voice range criteria for musical fitness are defined. Section 7.2 introduces the auditory attributes that are considered in this application. Section 7.3 explains the genetic encoding of notes and the evolutionary cycle for chord production. Section 7.4 describes the musical criteria applied as fitness functions. In Section 7.5 the graphical interface is presented, and the control bars are depicted. Section 7.6 discusses the experiments, and Section 7.7 presents the conclusions.

7.2 SOUND ATTRIBUTES

For our purposes, a sound, or an auditory event, is characterized by four parameters: pitch, timbre, loudness, and duration.

Pitch can be defined as the auditory property of a note that is conditioned by its frequency relative to the other notes. The range of musical pitch has been defined as the range within which the interval of an octave can be perceived. This has been found to correspond roughly to the range of the piano. From this continuum of frequencies, a set of discrete frequencies is selected in such a way that the frequencies bear a definite interval relationship among them. So, pitch in the musical sense corresponds to a frequency that is selected from a predefined repertoire. In this scheme, two discrete frequencies are chosen in the interval of an octave, so that the ratio between any two adjacent frequencies is $2^{1/12}$. In music terminology, this interval ratio is termed a *semitone.*

The other three parameters can be described straightforwardly: *timbre* is the individuality of sound acquired by the addition of harmonics to the fundamental pitch and is characteristic of a given musical instrument and the way of playing it. *Loudness* is the aspect of an auditory event related to its intensity, and, finally, *duration* is characterized by the period of time during which the event is perceivable.

Using these concepts, a *melody* is defined as a fixed temporal ordering of auditory events. In conventional occidental notation, a melody resembles a system of Cartesian coordinates. The pitch and duration are carefully marked; timbre is decided by the instrument for which it is written, and loudness is marked more crudely (Vidyamurthy and Chakrapani 1992).

The MIDI protocol provides a symbolic representation for the musical notes from which the above attributes may be extracted. Our approach uses MIDI events described by the MIDI note table for pitch representation, the MIDI velocity table for loudness and the general MIDI table for timbre. The duration is set in milliseconds.

7.3 EVOLUTIONARY MUSICAL CYCLE

In our approach, the population is made up of groups of four notes, which are potential solutions for a selection process. Genetic algorithms are used to generate and evaluate a sequence of chords. This sequence produces a sound result resembling a chord cadence or a fast counterpoint of note blocks. These are sent

1011111	1010111	0010111	0100111

7.1 FIGURE The structure of a MIDI chromosome.

to the MIDI port and can be heard as sound events in real time. Melodic, harmonic, and voice range fitnesses are used to control musical features. Based on the ordering of consonance of musical intervals, the concept of approximating a sequence of notes to its harmonically compatible note or tonal center is used. This method employs fuzzy formalism and is posed as an optimization approach based on factors relevant to hearing music, further described in Section 7.4.

7.3.1 The Voices Population

Individuals of the population are defined as groups of four voices, or notes. From now on, "voices" and "notes" will be used interchangeably. These voices are randomly generated in the interval [0..127], which corresponds to 7-bit values for a MIDI event. In each generation, 30 groups are generated. The group, or chord, is internally represented as a chromosome of 28 bits, or 4 words of 7 bits, one for each voice (Figure 7.1).

The following steps make up the genetic cycle (Pedrycz and Gomide 1998):

1. Create an initial population randomly.
2. While not stopped (by the user), perform the following:
   - Evaluate the musical fitness of each individual in the population.
   - Apply the genetic operators to the population of MIDI chromosomes (groups of voices), chosen based on musical fitness, to create a new population. The operators are *reproduction* (copy existing individual strings for a new population); *crossover* (create two new chromosomes by crossing over randomly chosen sublists (substrings) from two existing chromosomes); *mutation* (create new chromosomes from an existing one by randomly mutating a character in the list).
3. Find the best individual in the new population and play it as a MIDI event. Go to step 2.

The general architecture of the genetic cycle is depicted in Figure 7.2.

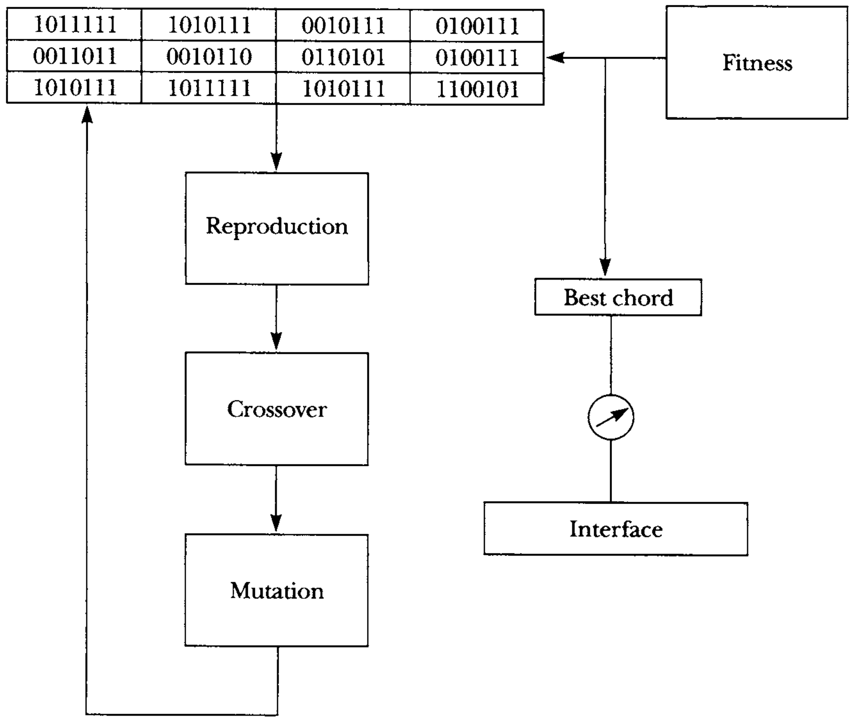


FIGURE 7.2 The genetic cycle.

7.3.2 The Rhythm of the Evolution

Musicians have always found inspiration in nature for their compositions. Water, fire, wind, rain, birds, and other natural sounds have appeared in their work. Prokofiev, in *Peter and the Wolf,* suggests both the sound of men and other animals. Earlier, *The Magic Flute,* by Mozart, explored the sounds of birds. Electroacoustic music uses the soundscape, quotidian noise, automobiles, hinges, and coins, in compositional context. Villa-Lobos explored several themes inspired by the Brazilian landscape. In *Bachiana No. 4,* he explored a train traveling along the countryside as a theme for developing his music. Sounds from the old Brazilian railroad were used by Raul do Valle and Jônatas Manzolli in a multimedia performance named "Trilhos Sonoros da Ferrovia" (Railroad Track Sound). A few years ago, in São Paulo, a group called The Sound Hunters roamed about the city recording the sounds of traffic, fountains, rush hour, and other everyday sounds, using them as samples in their compositions.

In a similar manner, inspired by machines with cyclical movements, such as trains or old mechanical sewing machines, we searched for a machine rhythm to express the Vox Populi evolution. What would be the rhythmic pattern of a computer when it is running a genetic algorithm? We decided to generate music with a rhythm built by the time necessary for selecting a better group of voices.

Two processes were integrated: (1) an evolving process generating individuals—groups of notes or voices, in this case—applying genetic operators and selecting individuals, and (2) the interface to look for notes to be played by the computer. When a best group is selected, it is put in a critical region that is continuously verified by the interface. These notes are played until the next group is selected. The timing of these two processes determines the rhythm of the music that is being heard. In any case, a graphical interface allows the user to interact with the rhythm, by modifying the cycles.

The steps above illustrate how many operations are executed in each cycle. The time interval between the selection of the best chords in two successive cycles may be different, while the interface is regularly "asking for new notes." Although the time taken to designate the best individual in each generation is approximately the same, small variations in each time cycle determine the *genetic rhythm;* the different times for the notes being played are perceived as the rhythm of the melody generated by the genetic cycle. The resulting system, Vox Populi, allows the user to modify the fitness function by means of four controls: The first is the melodic criterion; the second, the duration of the genetic cycle and musical rhythm; the third is the set of octave ranges to be considered; and the fourth, the time segment for each selected orchestra. All these controls are available for real-time performance, allowing the user to play and interact with Vox Populi's music evolution.

7.4 FITNESS EVALUATION

Since Western music is based on harmony, any general theory of music must address this matter. The term "harmony" is inherently ambiguous: It refers both to a lower level, where smoothness and roughness are evaluated, and to a higher aesthetic level, where harmony is functional in a given style. However, harmony is very subjective; the judgment of harmony does not seem to have a natural basis, but appears to be a common response acquired by people in a certain cultural setting. Therefore, opinions on the subject may vary widely depending on social and cultural backgrounds, and the many attempts to formalize the concept have proven inadequate. Nevertheless, while there is a difference of opinion on what constitutes harmony, there is a general agreement on the relative order of music interval consonance. Numerological theories of consonance have attempted to capture this aspect, but here again, a lot is left to the imagination, as the theory does not clearly delineate what constitutes the order of simplicity of music intervals.

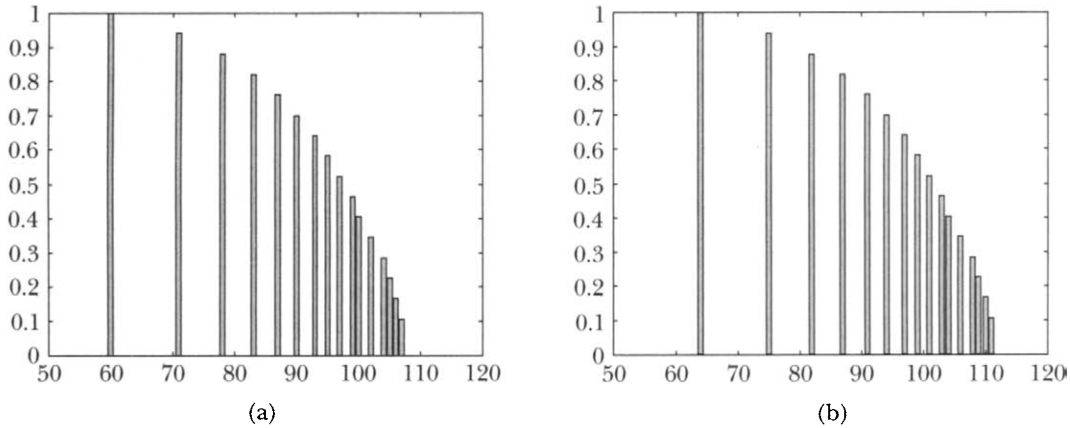
In our case, as a fitness function, a numerological theory of consonance from a physical point of view, based on a relative ordering of musical intervals, has been applied. Based on the ordering of consonance of these intervals, a sequence of notes is approximated to its harmonically compatible note or tonal center. Tonal centers can be thought of as an approximation of the melody that describes its flow. This method uses fuzzy formalism and is posed as an optimization problem based on physiological factors relevant to listening to music. This approach is significant because it does not adopt any heuristics.

7.4.1 The Consonance Criterion

Consonance is a combination of two simultaneous notes judged to be pleasing to the ear. In Vidyamurthy and Chakrapani (1992), harmony is introduced as a function of the commonality or the overlap of the harmonic series components of given fundamental notes. This overlap measurement is then scaled to a value between 0 and 1, with 1 denoting complete overlap (i.e., the two notes are the same) and 0 denoting no overlap at all. This concept of overlap can be succinctly captured in a fuzzy-set-based formalism.

A musical note is a compound tone consisting of its primary tone and upper harmonic series tones. It can be associated with a fuzzy set in which the degree of membership of each tone is proportional to its amplitude. Represented graphically as a spectrum, a musical note is a graph of frequency versus amplitude. To further enhance the musical comprehensibility of a note, the audible range frequency is mapped onto the keys of a piano. Using the chromatic scale, it is possible to make an approximation of this model using the piano keyboard. In what follows, the weighting of the note partials versus the harmonic series is presented, where n denotes the nth key on the piano, and $(n + k)$ denotes the key k semitones above key n. In Figure 7.3 the harmonic series for notes 60 and 64 are shown.

Formally, each note N is a fuzzy set S_N, comprising a collection of pairs (x, y), where x is a tone (also called a "partial"), and y is its "weight" in the note, which corresponds with the amplitude of that tone when the note is played. For example, the set S_{60} contains 16 such pairs and is illustrated by the graph in Figure 7.3 (a), which plots the points (x, y). When we thus represent a note as a fuzzy set, the y values are normalized so that the total weights of all partials sum to 1.



7.3 FIGURE The upper partials for two notes: (a) 60 and (b) 64.

If S_N is a note, and U is the set of all tones, we therefore have

$$S_N = \{(x,y) \mid x \in U, y \in [0,1]\}, \text{ where } \sum_{y \in \{y \mid (x,y) \in S_N\}} y = 1$$

Given a pair of notes S_M, S_N, we can define their *intersection* $S_{M \cap N}$, as follows:

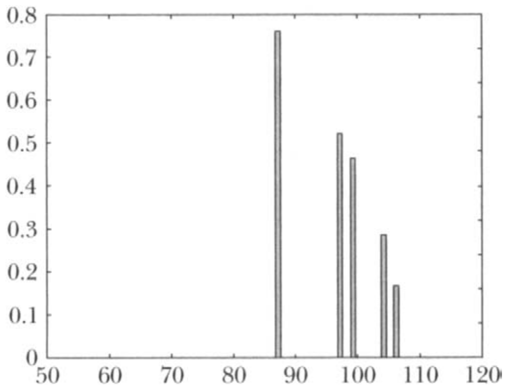
$$S_{M \cap N} = \{(x,y) \mid (x,a) \in S_M \wedge (x,b) \in S_N \wedge y = \min\{a,b\}\}$$

That is, the intersection is a collection of partials involving only the tones represented in both notes, and with each such tone given the lowest of the weights it has in the notes themselves. For example, Figure 7.4 depicts the intersection of the two notes shown in Figure 7.3.

We can now define the consonance, or overlap, of a pair of notes S_M, S_N, as follows:

$$C_O(S_M, S_N) = \sum_{(x,y) \in S_{M \cap N}} y$$

Simply stated, the consonance of a pair of notes may be interpreted as the sum of the intersection of the partial weights, with results in the range [0,1].



7.4 FIGURE The weighting of the *intersection* of notes 60 and 64.

7.4.2 Melodic Fitness

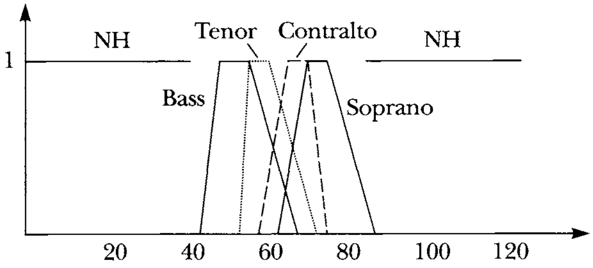
Melody is defined as an ordered sequence of notes with their corresponding start times. In formal terms, a melody can be a string in which each character is an ordered pair (S_N, t), where S_N is the note and t is its duration. Given an ordered sequence of notes, it seems intuitively appealing to call the note that is most consonant with all the other notes, the "coloring" or "tonal center." Hence, the extraction of the tonal center of a sequence of notes would involve finding a single harmonically compatible note, such that the time weighted dissonance between a given note and any other one in the sequence is minimized. In Vox Populi, this is measured according to the weighting value Id, which is obtained from the interface control and can be varied by the user. To derive the melodic fitness of a string of four notes $(x_1,t_1),(x_2,t_2),(x_3,t_3),(x_4,t_4)$, we therefore first derive their tonal center note Id and calculate

$$M(x_1,x_2,x_3,x_4) = \max_{i=1,\ldots,4}\{Co(x_i,Id)\}$$

7.4.3 Harmonic Fitness

Harmony is defined here as a function of the commonality or overlap of the spectral components of the notes and the sum of the two-by-two consonance. Therefore, the harmonic fitness is defined as

$$H(x_1,x_2,x_3,x_4) = Co(S_{x_1},S_{x_2}) + Co(S_{x_2},S_{x_3}) + Co(S_{x_3},S_{x_4}) + \mathrm{Co}(S_{x_1},S_{x_4})$$



7.5 FIGURE The linguistic values associated with the voices.

7.4.4 Voice Range Criterion

Voices are associated with the linguistic terms "bass," "tenor," "contralto," "soprano," and "nonhuman" (NH). The related fuzzy sets are shown in Figure 7.5, where the units on the horizontal axis are MIDI note values. Each voice is assigned to a membership value associated with each linguistic term in the set {NH, B, T, C, S}. For example, the MIDI note value 60 has the degrees of membership 0.5, 1.0, and 0.5 in the bass, tenor, and contralto sets, respectively. Given a note, the membership of that note in each set is calculated, and the maximum value is taken to indicate the voice of that note. In case of a tie, the voice classification is based on the distance of that note from the centers of the two fuzzy sets concerned.

Once the voices of each group are evaluated according to their membership in the interval of voices, the voice range criterion returns the maximum in each interval. Therefore, the voice range fitness is evaluated as

$$O(x_1, x_2, x_3, x_4) = \left(\sum_{i \in \{1,\ldots,4\}} V(x_i) \right) / 4$$

where $V(x_i)$ is the membership value of note x_i associated with its assigned voice.

7.4.5 Musical Fitness

The resulting musical fitness is a conjunction of the previous functions and is defined as

$$F(O,M,H) = O(x_1,x_2,x_3,x_4) + M(x_1,x_2,x_3,x_4) + H(x_1,x_2,x_3,x_4)$$

In the selection process, the group of voices with the highest fitness is selected and played.

7.5 INTERFACE AND PARAMETER CONTROL

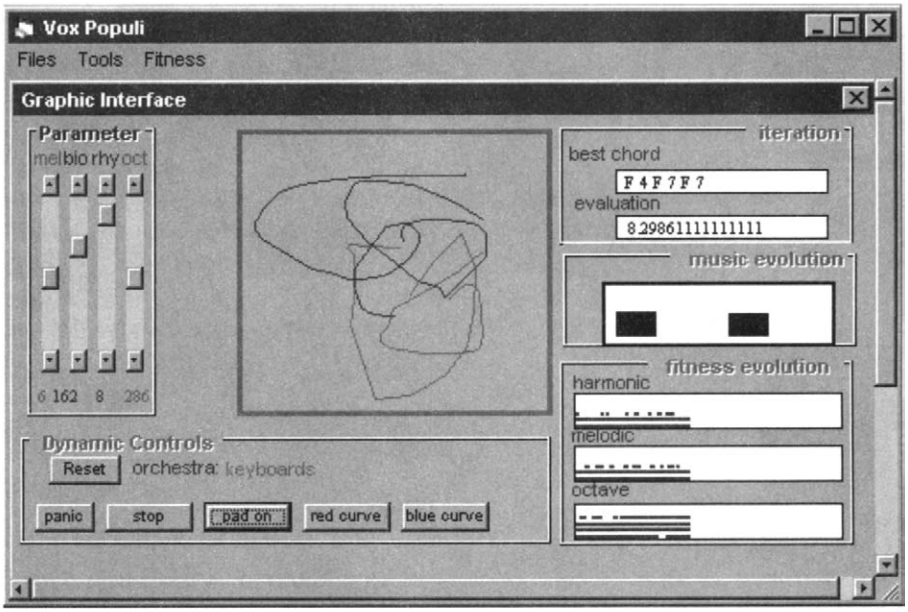
The system was designed to perform a series of sound experiments and to be used as a tool for algorithmic composition, with satisfactory performance, the ability to interact with sound, and a possible use of Vox Populi for live music. The program was developed for the MS Windows environment and runs on any personal computer and most commercial soundboards. The graphical interface allows the user to change parameters and interfere with the fitness function and, consequently, with the musical evolution.

The resulting music moves from very pontilistic sounds to sustained chords; it depends on the duration of the genetic cycle and the number of individuals in the original population. The octave fitness forces the notes to be in the range *H*, assumed to be the range reached by the human voice and associated with the center region of the notes on the piano, but since several orchestras of instruments are used, this range is too limited for some of them. The original decision to restrict the generated voices to specific ranges was only to resemble the voices to human voices; nevertheless, a user can enlarge these ranges by using the octave control. The interface was designed to be flexible so that the user can modify the music being generated.

Next we present a short description of the controls available for user interaction with Vox Populi. The descriptions should be read with reference to Figure 7.6, which shows Vox Populi's graphical interface.

Melodic control: The Mel scroll bar allows the user to modify the value of *Id*, the tonal center in the evaluation of the melodic fitness.

Biological control: The Bio scroll bar allows the user to interfere with the duration of the genetic cycle, modifying the time between genetic iterations. Since the music is being generated in real time, this artifice was necessary for the timing of the different processes that are running. This value determines the slice of time necessary to apply the genetic operators, such as crossover and mutation, and can also be interpreted as the reproduction time for each generation.



7.6 FIGURE Vox Populi's graphical interface.

Rhythmic control: The Rhy scroll bar changes the time between evaluations of the musical fitness. It determines the "time to produce a new generation," or the slice of time necessary to evaluate the musical fitness of the population. It interferes directly with the musical rhythm. Changes in this control make the rhythm faster or slower.

Octave control: The Oct scroll bar allows the interval of voices considered in the voice range criterion to be enlarged or decreased.

Orchestra control: Six MIDI orchestras are used to play the selected chords: (1) keyboards; (2) strings and brasses; (3) keyboards, strings, and percussion; (4) percussion; (5) sound effects; and (6) a random orchestra, by taking an instrument from the general MIDI list. Using the order above, these orchestras are sequentially changed in time segments controlled by the Seg scroll bar.

Interactive pad control: The Pad On button enables and disables the pad's effect on the controls defined above. These controls can be coupled in two pairs, which can be interpreted as variables of a two-dimensional phase space, allowing a user to draw an oriented curve to determine the musical evolution. There are two curves, each associated with a different color. The red curve describes a phase space of the *melodic* and *octave range* control variables. The blue curve describes a phase space of the *biological* and *rhythmic* control variables.

The pad may be musically interpreted as an elementary tool that allows a "master gesture" to conduct the music.

Fitness displays: Three other displays allow the user to follow the fitness evolution. The first display, on the top right, shows the notes and the fitness of the chord that is being played. In the middle, a bar graph shows the four voices (bass, tenor, contralto, soprano) and their value. It is equivalent to the membership function values related to the range of the voices. The last display shows the melodic, harmonic, and octave fitness bars.

7.6 EXPERIMENTS

Two main approaches were tried to express the fitness evaluation, and both presented interesting results. The first one, derived from a heuristic model, provides a less expensive fitness evaluation. The disadvantage of this approach is that it demands a strong musical background to formulate and understand it. Because of this, its presentation will be omitted. This method allows the use of a large population, between 100 and 200 chords, producing more diversification and resulting in a slow convergence to the best chord sequence.

In the second approach, the consonance criterion presented above is used, and a longer calculation is needed to evaluate the music fitness. In order to assure good real-time performance, the population is limited to 30 chords. Since the musical fitness utilized was tighter, the resulting sound output was less diversified, and it was possible to hear the musical sequence converging to unison. This fact highlights the idea that in musical composition not only is the consonance desirable, but also the dissonance.

The octave control offers an interesting tool to vary the voice performance, diminishing or enlarging the voice range. The resulting sounds are isolated note blocks or chords. The larger the range, the larger the chord separation. This control highlights the automated nature of the Vox Populi very well since it produces some combinations of notes that cannot be played by a human musician because of physical limitations (distance among the notes and speed, for example). Another feature that has been added to the system is a music performance menu (Figure 7.7), which offers the possibility of playing solos, chords, or arpeggios.

FIGURE 7.7 Performance menu.

Using this entire range of musical features, impromptu musical sessions were held with live musicians, one controlling Vox Populi and the other accompanying him with a synthesizer keyboard. As a musical partner, Vox Populi can produce interesting rhythmic patterns, chord cadences, or pontilistic sound sequences for long periods. Sometimes the music seemed slightly predictable, but when the pad control was used to vary Vox Populi's parameters, the music changed to unexpected cadences, then converged to another pointillistic sequence, and so on. Maybe the best expression to describe this dialogue between the system and live musicians would be "spontaneity."

7.7 CONCLUSIONS

Vox Populi uses the computer and the mouse as real-time music controllers, producing dynamic musical structures based on evolutionary models. It is a new interactive computer-based musical instrument. It explores evolutionary computation in the context of algorithmic composition and uses a graphical interface to change the musical evolution. These results reflect current concerns at the forefront of computer music: interactive composition and the development of new controller interfaces. A demo of Vox Populi is available on the CD-ROM and at *http://www.ia.cti.br/~artemis/voxpopuli.*

The use of Vox Populi as a compositional tool in a real-time algorithmic composition environment at the electronic music studio of NICS (Interdisciplinary Nucleus of Sound Communication) showed the potential of this system to control the evolution of music. As a representation of a musical creative process, this system was integrated to interpretative improvisations created by musicians producing pleasant musical cadences. This interactivity emphasizes aspects of musical practices in the scope of human/machine interaction.

Furthermore, we have been able to integrate this system with gesture interfaces, such as gloves, to enhance the human/machine interaction, with the goal of allowing a human gesture to be the real-time controller. This is a natural extension of the pad control. This approach was previously applied to the design of ActContAct (Manzolli, Moroni, and Matallo 1998), where an electronic shoe was used for music generation. In this way, Vox Populi, designed to reflect the natural evolution of the sound domain, will return its control to the human agent during its own life cycle. It would certainly be interesting to follow up on the use of computers to develop new ways of human interaction through music.

ACKNOWLEDGMENTS

We would like to thank our fellow student Leonardo N. S. Pereira for developing the routines to evaluate the consonance criterion. This work was supported by FAPESP (São Paulo State Research Foundation) and CTI (Technological Center for Informatics), and CNPq (process no. 300910/96–7).

REFERENCES

Apel, W. (1958). *Gregorian Chant.* Indiana University Press.

Bidlack, R. (1992). Chaotic Systems as Simple (but Complex) Compositional Algorithms. *Computer Music Journal* 16(3):33–42.

Hiller, L., and R. Baker (1964). Computer Cantata: A Study of Compositional Method. *Perspectives of New Music* 3(1).

Horovitz, D. (1994). Generating Rhythms with Genetic Algorithms. In *Proceedings of the International Computer Music Conference* (*ICMC '94*), pp. 142–143.

Loy, G. (1988). Composing with Computers—A Survey of Some Compositional Formalism and Music Programming Languages. In M. V. Matthews and J. R. Pierce (eds.), *Current Directions in Computer Music Research,* MIT Press, pp. 291–396.

Manzolli, J. (1993). *Non-linear Linear Dynamics and Fractals as a Model for Sound Synthesis and Real Time Composition.* Ph.D. dissertation, University of Nottingham, UK.

Manzolli, J., A. Moroni, and C. Matallo (1998). AtoConAto: New Media Performance for Video and Interactive Tap Shoes Music. In *Proceedings of the 6th ACM International Multimedia Conference.*

Mathews, M. (1963). The Digital Computer as a Musical Instrument. *Science* 142.

Miranda, E. R. (1994). Music Composition Using Cellular Automata. *Languages of Design* 2:105–117.

Moore, R. F. (1990). *Elements of Computer Music.* Prentice-Hall.

Pedrycz, W., and F. Gomide (1998). *An Introduction to Fuzzy Sets Analysis and Design.* MIT Press.

Strunk, O. (1950) *Source Readings in Music History.* Vail-Ballou Press.

Vidyamurthy, G., and J. Chakrapani (1992). Cognition of Tonal Centers: A Fuzzy Approach. *Computer Music Journal* 16(2).